Benchmarking the (1,4)-CMA-ES With Mirrored Sampling and Sequential Selection on the Noiseless BBOB-2010 Testbed

[Black-Box Optimization Benchmarking Workshop]

Anne Auger, Dimo Brockhoff, and Nikolaus Hansen
Projet TAO, INRIA Saclay—Ile-de-France
LRI, Bât 490, Univ. Paris-Sud
91405 Orsay Cedex, France
firstname.lastname@inria.fr

ABSTRACT

The well-known Covariance Matrix Adaptation Evolution Strategy (CMA-ES) is a robust stochastic search algorithm for optimizing functions defined on a continuous search space $\mathbb{R}^D$. Recently, mirrored samples and sequential selection have been introduced within CMA-ES to improve its local search performances. In this paper, we benchmark the (1,$4_{\rm m}^{\rm s}$)-CMA-ES which implements mirrored samples and sequential selection on the BBOB-2010 noiseless testbed. Independent restarts are conducted until a maximal number of $10^4 D$ function evaluations is reached.

The experiments show that 11 of the 24 functions are solved in 20D (and 13 in 5D respectively). Compared to the function-wise target-wise best algorithm of the BBOB-2009 benchmarking, on 25% of the functions the (1,$4_{\rm m}^{\rm s}$)-CMA-ES is at most by a factor of 3.1 (and 3.8) slower in dimension 20 (and 5) for targets associated to budgets larger than $10D$. Moreover, the (1,$4_{\rm m}^{\rm s}$)-CMA-ES slightly outperforms the best algorithm on the rotated ellipsoid function in 20D and would be ranked two on the Gallagher function with 101 peaks in 10D and 20D—being 25 times faster than the BIPOP-CMA-ES and about 3 times faster than the (1+1)-CMA-ES on this function.

Categories and Subject Descriptors

G.1.6 [**Numerical Analysis**]: Optimization—*global optimization, unconstrained optimization*; F.2.1 [**Analysis of Algorithms and Problem Complexity**]: Numerical Algorithms and Problems

General Terms

Algorithms

Keywords

Benchmarking, Black-box optimization

1. INTRODUCTION

Evolution Strategies (ESs) are stochastic search algorithms designed to minimize[1] objective functions, f, mapping a continuous search space $\mathbb{R}^D$ into $\mathbb{R}$. Among ESs, the Covariance Matrix Adaptation ES (CMA-ES) is now a well-recognized algorithm. In the standard $(\mu/\mu_w, \lambda)$-CMA-ES [17, 23], at each iteration step n, a set of λ candidate solutions is created by sampling random vectors distributed according to a multivariate normal distribution with mean vector zero and covariance matrix $\boldsymbol{C}_n$. Those λ random vectors denoted $(\mathcal{N}_i(\mathbf{0}, \boldsymbol{C}_n))_{1 \leq i \leq \lambda}$ are multiplied by a strictly positive factor, the step-size σ_n, and added to the current solution $\boldsymbol{X}_n$ to constitute the offspring $\boldsymbol{X}_n^i = \boldsymbol{X}_n + \sigma_n \mathcal{N}_i(\mathbf{0}, \boldsymbol{C}_n)$. After evaluation of the λ offspring, the μ best, i.e., the ones having the smallest objective function values, are selected. The current solution is updated to the average value of the μ best solutions: $\boldsymbol{X}_{n+1} = \sum_{i=1}^{\mu} w_i \boldsymbol{X}_n^{i:\lambda}$, where $w_1 \geq \ldots \geq w_\mu$ and $\sum_{i=1}^{\mu} w_i = 1$ and $\boldsymbol{X}_n^{i:\lambda}$ denotes the i-th best offspring. Covariance matrix and step-size are then updated using solely the information given by the ranking of the offspring. Though originally designed to be a robust local search [24], the $(\mu/\mu_w, \lambda)$-CMA-ES turns out to be also effective for multi-modal functions provided a large enough population size $\mu = \lambda/2$ is chosen [23]. An automatic way to increase the probability to converge on multi-modal functions consists in applying restarts with a successively increasing population size. The strategy is then called IPOP-CMA-ES [12]. However, deceptive functions were constructed for the IPOP-CMA-ES [25, 21]. The BBOB function f_{24} presents, in a highly rugged landscape, on the larger scale an attraction region for the global optimum which is smaller than the one for the local optimum. For that reason, the BIPOP-CMA-ES, combining restarts with increasing population size as well as with some small population size, was proposed [18]. For the large budgets that are needed for most multi-modal problems, the BIPOP-CMA-ES performed overall best in the BBOB-2009 benchmarking [20].

While BIPOP-CMA-ES was benchmarked, the local search

[1]We assume without loss of generality minimization since maximizing f amounts to minimize $-f$.

(1+1)-CMA-ES was as well tested [13, 14]. Surprisingly, the (1+1)-variant of CMA-ES could outperform the BIPOP-CMA-ES algorithm by a significant factor on the Gallagher functions f_{21} and f_{22} [10]. On f_{21}, the (1+1)-CMA-ES is 8.2 times (resp. 68.7 times) faster than the BIPOP-CMA-ES in dimension 20 (resp. 40); for f_{22}, the (1+1)-CMA-ES is 37 times faster than the BIPOP-CMA-ES in 20D and is able to solve the problem in 40D which the BIPOP-CMA-ES does not allow. However, one major drawback of elitist selection, used in the (1+1)-CMA-ES, is the complete lack of robustness in presence of noise [14].

Motivated by the surprisingly large improvement over the BIPOP-CMA, new *non-elitist* local search ESs have been proposed [5]. Those $(1,\lambda)$-ESs combine a derandomization technique by means of *mirrored samples* with a *sequential selection* scheme. Mirrored samples replace the independent random vectors used for the offspring. Instead of the λ independent random vectors, only $\lambda/2$ (assuming λ is even) independent samples are generated $(\mathcal{N}_{2i-1}(\mathbf{0},\boldsymbol{C}_n))_{1\leq i\leq\lambda/2}$. The other $\lambda/2$ samples are replaced by the already generated samples multiplied by -1, i.e., $\mathcal{N}_{2i}(\mathbf{0},\boldsymbol{C}_n) = -\mathcal{N}_{2i-1}(\mathbf{0},\boldsymbol{C}_n)$ for all $1 \leq i \leq \lambda/2$. The resulting offspring are two by two symmetrical or *mirrored* with respect to $\boldsymbol{X}_n$. Sequential selection consists in performing the evaluations of the λ offspring sequentially and comparing after each evaluation the offspring solution $\boldsymbol{X}_n^i$ with the current solution $\boldsymbol{X}_n$. If $f(\boldsymbol{X}_n^i) \leq f(\boldsymbol{X}_n)$, the sequence of evaluations is stopped and $\boldsymbol{X}_{n+1} = \boldsymbol{X}_n^i$—saving the remaining offspring evaluations.

The impact of mirrored samples and sequential selection has been investigated on the BBOB-2010 for the (1,2)-CMA-ES [1, 2, 6, 7] and for the (1,4)-CMA-ES [3, 4, 8, 9]. The purpose of this paper is to present the results of one of those strategies tested, namely the (1,4)-CMA-ES with mirrored samples and sequential selection on the BBOB-2010 noiseless testbed. Since the algorithm tested is a local-search strategy, we do not expect that it will perform well on the whole testbed but rather want to see whether the strategy can bring improvements over last year's results on *certain* functions.

2. THE (1,$4^{\mathrm{s}}_{\mathrm{m}}$)-CMA-ES

Compared to the $(\mu/\mu_w,\lambda)$-CMA-ES sketched in the introduction, the (1,$4^{\mathrm{s}}_{\mathrm{m}}$)-CMA-ES selects a single offspring, i.e., $\mu = 1$ out of $\lambda = 4$. Moreover, mirrored samples are applied such that $\boldsymbol{X}_n^i = \boldsymbol{X}_n + \sigma_n\mathcal{N}_i(\mathbf{0},\boldsymbol{C}_n)$ for $i = 1,3$ and $\boldsymbol{X}_n^i = \boldsymbol{X}_n - \sigma_n\mathcal{N}_{i-1}(\mathbf{0},\boldsymbol{C}_n)$ for $i = 2,4$. Thus, $\boldsymbol{X}_n^1$ and $\boldsymbol{X}_n^2$ ($\boldsymbol{X}_n^3$ and $\boldsymbol{X}_n^4$) are symmetric with respect to $\boldsymbol{X}_n$. In addition, sequential selection is applied. Evaluations are carried out in a sequential manner, i.e., after evaluating the ith offspring solution $\boldsymbol{X}_n^i$, it is compared to $\boldsymbol{X}_n$ and if $f(\boldsymbol{X}_n^i) \leq f(\boldsymbol{X}_n)$, the sequence of evaluations is concluded and $\boldsymbol{X}_{n+1} = \boldsymbol{X}_n^i$. In case the four offspring solutions are worse than $\boldsymbol{X}_n$, $\boldsymbol{X}_{n+1} = \operatorname{argmin}\{f(\boldsymbol{X}_n^1), \ldots, f(\boldsymbol{X}_n^4)\}$ according to the comma selection. Note that the number of offspring evaluated is a random variable by itself ranging from 1 to $\lambda = 4$—allowing to reduce the number of offspring adaptively.

2.1 Independent Restarts

Similar to [11], we independently restarted (1,$4^{\mathrm{s}}_{\mathrm{m}}$)-CMA-ES as long as function evaluations were left, where $10^4 \cdot D$ has been used as the maximal number of function evaluations.

2.2 Parameter setting

We used the default parameter and termination settings (cf. [5, 18, 23]) found in the source code on the WWW[2] with two exceptions. We rectified the learning rate of the rank-one update of the covariance matrix for small values of λ, setting $c_1 = \min(2,\lambda/3)/((D+1.3)^2 + \mu_{\mathrm{eff}})$. The original value was not designed to work for $\lambda < 5$. We modified the damping parameter for the step-size to $d_\sigma = 0.3 + 2\mu_{\mathrm{eff}}/\lambda + c_\sigma$. The setting was found by performing experiments on the sphere function, f_1: d_σ was set as large as possible while still showing close to optimal performance, but, at least as large such that decreasing it by a factor of two did not lead to inacceptable performance. For $\mu_{\mathrm{eff}}/\lambda = 0.35$ and $\mu_{\mathrm{eff}} \leq D+2$ the former setting of d_σ is recovered. For a smaller ratio of $\mu_{\mathrm{eff}}/\lambda$ or for $\mu_{\mathrm{eff}} > D+2$, the new setting allows larger (i.e. faster) changes of σ. Here, $\mu_{\mathrm{eff}} = 1$. For $\lambda \geq 3$, the new setting might be harmful in a noisy or too rugged landscape. Finally, the step-size multiplier was clamped from above at $\exp(1)$, while we do not believe this had any effect in the presented experiments. Each initial solution $\boldsymbol{X}_0$ was uniformly sampled in $[-4,4]^D$ and the step-size σ_0 was initialized to 2. The source code used for the experiments is available at[3].

3. CPU TIMING EXPERIMENTS

For the timing experiment, (1,$4^{\mathrm{s}}_{\mathrm{m}}$)-CMA-ES was run on f_8 with a maximum of $10^4 D$ function evaluations and restarted until at least 30 seconds have passed (according to Figure 2 in [19]). The experiments have been conducted with an 8 core Intel Xeon E5520 machine with 2.27 GHz under Ubuntu 9.1 linux and Matlab R2008a. The time per function evaluation was 7.1; 7.3; 7.7; 8.1; 7.1; 8.1 times 10^{-4} seconds in dimensions 2; 3; 5; 10; 20; 40 respectively. Note that MATLAB distributes the computations over all 8 cores only for 20D and 40D.

4. RESULTS AND DISCUSSION

Results from experiments according to [19] on the benchmark functions given in [16, 22] are presented in Figures 1, 2 and 3 and in Tables 1 and 2.

Overall, we can state that 11 (respectively 13) of the 24 functions are solved in 20D (in 5D). With the exception of the Gallagher functions f_{21} and f_{22}, none of the multi-modal and weakly-structured problems f_{15-24} has been solved. Table 2 shows that the (1,$4^{\mathrm{s}}_{\mathrm{m}}$)-CMA-ES is finally on half of the functions by a factor of at most 43 (respectively 10) slower in 20D (in 5D) than the function-wise best algorithm of the BBOB-2009 benchmarking.

The comparison with the function-wise best algorithm of BBOB-2009 is even more interesting on specific functions. On the rotated ellipsoid (f_{10}) in 20D, for example, the (1,$4^{\mathrm{s}}_{\mathrm{m}}$)-CMA-ES has a slightly lower expected running time (about 2%) than the best algorithm of BBOB-2009 on this function [10], which is the (1+1)-CMA-ES [13] for a target value of 10^{-7} and the iAMaLGaM IDEA [15] for a target of 10^{-5}. On the Gallagher function with 101 peaks (f_{21}) in 20D, the (1,$4^{\mathrm{s}}_{\mathrm{m}}$)-CMA-ES is only slightly worse (factors 1.68, 1.65, and 1.47 for target values of 10^{-4}, 10^{-5},

[2] `cmaes.m`, version `3.41.beta`, from `http://www.lri.fr/~hansen/cmaes_inmatlab.html`

[3] `http://coco.gforge.inria.fr/doku.php?id=bbob-2010-results`

Table 2: ERT loss ratio (see Figure 3) compared to the respective best result from BBOB-2009 for budgets given in the first column. The last row $\mathrm{RL_{US}}$/D gives the number of function evaluations in unsuccessful runs divided by dimension. Shown are the smallest, 10%-ile, 25%-ile, 50%-ile, 75%-ile and 90%-ile value (smaller values are better). Data presented correspond to the data from the upper plots in Figure 3.

	f_1–f_{24} in 5-D, maxFE/D=10000					
#FEs/D	best	10%	25%	**med**	75%	90%
2	1.7	1.9	2.4	3.3	5.9	8.5
10	1.2	1.6	2.0	3.1	4.1	20
100	0.95	2.3	3.4	6.3	9.5	30
1e3	0.69	1.9	3.8	5.7	13	27
1e4	0.69	1.9	3.8	9.9	28	46
$\mathrm{RL_{US}}$/D	1e4	1e4	1e4	1e4	1e4	1e4

	f_1–f_{24} in 20-D, maxFE/D=10000					
#FEs/D	best	10%	25%	**med**	75%	90%
2	0.83	1.2	8.1	37	40	40
10	2.5	3.1	3.8	4.9	45	2.0e2
100	1.4	2.4	3.1	7.4	17	74
1e3	0.98	1.2	3.1	8.6	53	1.6e2
1e4	0.98	1.2	3.1	37	1.4e2	3.5e2
1e5	0.98	1.2	3.1	43	7.9e2	2.3e3
$\mathrm{RL_{US}}$/D	1e4	1e4	1e4	1e4	1e4	1e4

and 10^{-7} respectively) than the best algorithms of BBOB-2009 on this function, NEWUOA [27] and GLOBAL [26]—becoming ranked two when compared to all results of BBOB-2009 and being 25 times faster than the BIPOP-CMA-ES [18] and about 3 times faster than the (1+1)-CMA-ES [13] on this function. Similar results on f_{21} hold for 10D, where GLOBAL is the only best algorithm for all small target values. For the second Gallagher function with 21 peaks (f_{22}), the (1,$4_{\mathrm{m}}^{\mathrm{s}}$)-CMA-ES is by a factor of 5 worse than the best algorithm of BBOB-2009 on this function in 20D and a target function value of 10^{-7}, i.e., the (1+1)-CMA-ES, but still by a factor of more than 7 better than the BIPOP-CMA-ES. Note that for the function f_{22}, only 4 runs were successful such that we expect to get more reliable results if we run the algorithm a bit longer than the $10^4 D$ function evaluations. A similar statement holds for the sharp ridge f_{13} where only two runs successfully reached the target of 10^{-5}.

Acknowledgments

This work receives support by the French national research agency (ANR) within the SYSCOMM project ANR-08-SYSC-017 and within the COSINUS project ANR-08-COSI-007-12.

5. REFERENCES

[1] A. Auger, D. Brockhoff, and N. Hansen. Investigating the impact of sequential selection in the (1,2)-CMA-ES on the noiseless BBOB-2010 testbed. In *GECCO (Companion)*, 2010.

[2] A. Auger, D. Brockhoff, and N. Hansen. Investigating the impact of sequential selection in the (1,2)-CMA-ES on the noisy BBOB-2010 testbed. In *GECCO (Companion)*, 2010.

[3] A. Auger, D. Brockhoff, and N. Hansen. Investigating the impact of sequential selection in the

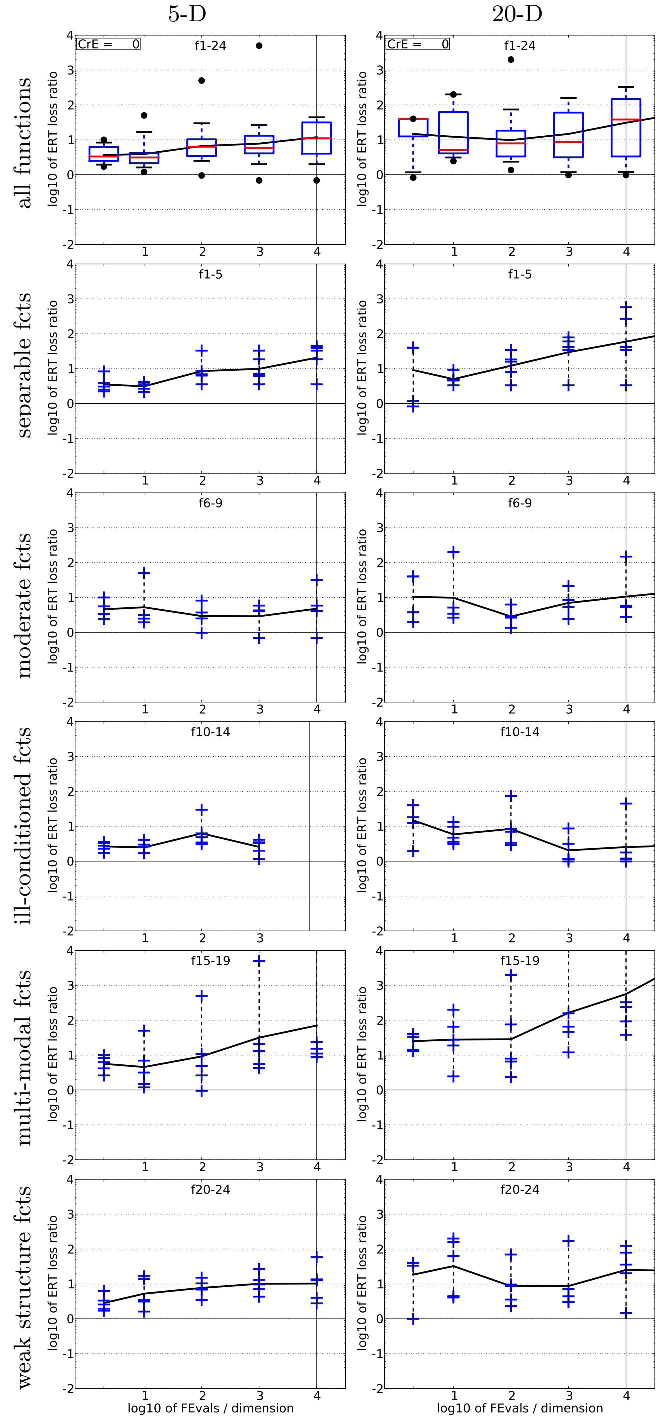

Figure 3: ERT loss ratio versus given budget FEvals. The target value f_{t} for ERT (see Figure 1) is the smallest (best) recorded function value such that $\mathrm{ERT}(f_{\mathrm{t}}) \leq \mathrm{FEvals}$ for the presented algorithm. Shown is FEvals divided by the respective best $\mathrm{ERT}(f_{\mathrm{t}})$ from BBOB-2009 for functions f_1–f_{24} in 5-D and 20-D. Each ERT is multiplied by $\exp(\mathrm{CrE})$ correcting for the parameter crafting effort. Line: geometric mean. Box-Whisker error bar: 25-75%-ile with median (box), 10-90%-ile (caps), and minimum and maximum ERT loss ratio (points). The vertical line gives the maximal number of function evaluations in this function subset.

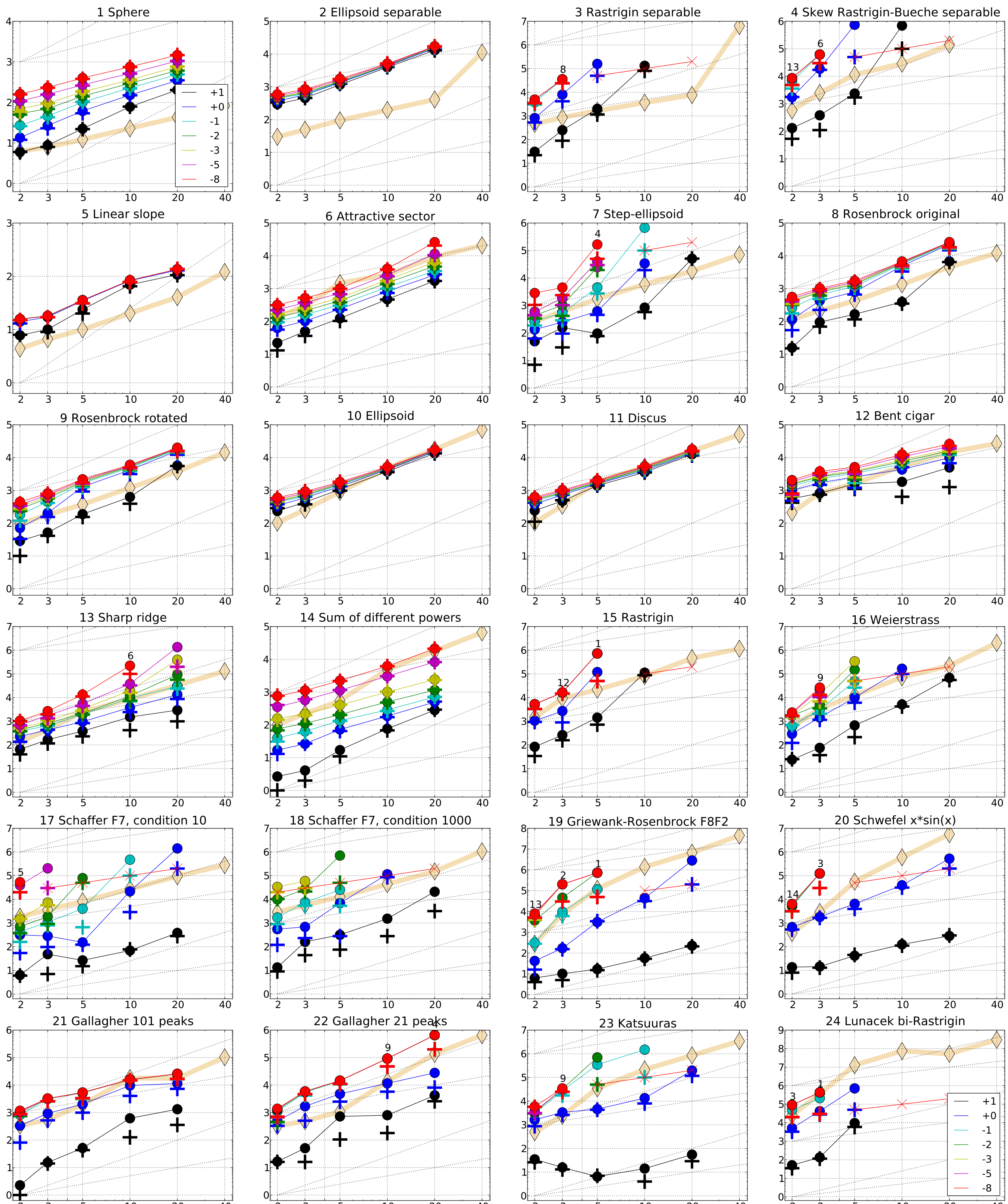

Figure 1: Expected Running Time (ERT, ●) to reach $f_{\mathrm{opt}} + \Delta f$ and median number of f-evaluations from successful trials (+), for $\Delta f = 10^{\{+1,0,-1,-2,-3,-5,-8\}}$ (the exponent is given in the legend of f_1 and f_{24}) versus dimension in log-log presentation. For each function and dimension, $\mathrm{ERT}(\Delta f)$ equals to $\#\mathrm{FEs}(\Delta f)$ divided by the number of successful trials, where a trial is successful if $f_{\mathrm{opt}} + \Delta f$ was surpassed. The $\#\mathrm{FEs}(\Delta f)$ are the total number (sum) of f-evaluations while $f_{\mathrm{opt}} + \Delta f$ was not surpassed in the trial, from all (successful and unsuccessful) trials, and f_{opt} is the optimal function value. Crosses (×) indicate the total number of f-evaluations, $\#\mathrm{FEs}(-\infty)$, divided by the number of trials. Numbers above ERT-symbols indicate the number of successful trials. Y-axis annotations are decimal logarithms. The thick light line with diamonds shows the single best results from BBOB-2009 for $\Delta f = 10^{-8}$. Additional grid lines show linear and quadratic scaling.

Δf	**f_{1} in 5-D**, N=15, mFE=508					**f_{1} in 20-D**, N=15, mFE=1720				
	#	ERT	10%	90%	RT_{succ}	#	ERT	10%	90%	RT_{succ}
10	15	2.3e1	1.1e1	4.5e1	2.3e1	15	2.0e2	1.6e2	2.4e2	2.0e2
1	15	6.2e1	3.6e1	8.8e1	6.2e1	15	3.5e2	2.9e2	4.0e2	3.5e2
1e−1	15	1.0e2	7.1e1	1.3e2	1.0e2	15	4.9e2	4.3e2	5.3e2	4.9e2
1e−3	15	1.8e2	1.3e2	2.4e2	1.8e2	15	7.6e2	6.7e2	8.5e2	7.6e2
1e−5	15	2.7e2	2.2e2	3.5e2	2.7e2	15	1.1e3	9.3e2	1.1e3	1.1e3
1e−8	15	4.0e2	3.5e2	4.9e2	4.0e2	15	1.5e3	1.3e3	1.6e3	1.5e3

Δf	**f_{2} in 5-D**, N=15, mFE=2006					**f_{2} in 20-D**, N=15, mFE=18752				
	#	ERT	10%	90%	RT_{succ}	#	ERT	10%	90%	RT_{succ}
10	15	1.2e3	7.6e2	1.5e3	1.2e3	15	1.3e4	1.2e4	1.5e4	1.3e4
1	15	1.3e3	8.8e2	1.5e3	1.3e3	15	1.5e4	1.3e4	1.6e4	1.5e4
1e−1	15	1.4e3	1.2e3	1.6e3	1.4e3	15	1.6e4	1.5e4	1.6e4	1.6e4
1e−3	15	1.5e3	1.3e3	1.7e3	1.5e3	15	1.6e4	1.6e4	1.7e4	1.6e4
1e−5	15	1.6e3	1.5e3	1.8e3	1.6e3	15	1.7e4	1.6e4	1.7e4	1.7e4
1e−8	15	1.8e3	1.6e3	2.0e3	1.8e3	15	1.7e4	1.7e4	1.8e4	1.7e4

Δf	**f_{3} in 5-D**, N=15, mFE=50004					**f_{3} in 20-D**, N=15, mFE=200004				
	#	ERT	10%	90%	RT_{succ}	#	ERT	10%	90%	RT_{succ}
10	15	2.0e3	2.0e2	6.3e3	2.0e3	0	*61e+0*	*39e+0*	*67e+0*	1.3e5
1	4	1.6e5	1.8e4	3.9e5	2.1e4	.	.	.	.	.
1e−1	0	*20e−1*	*99e−2*	*30e−1*	2.0e4	.	.	.	.	.
1e−3	.	.	.	.	.	.	.	.	.	.
1e−5	.	.	.	.	.	.	.	.	.	.
1e−8	.	.	.	.	.	.	.	.	.	.

Δf	**f_{4} in 5-D**, N=15, mFE=50004					**f_{4} in 20-D**, N=15, mFE=200004				
	#	ERT	10%	90%	RT_{succ}	#	ERT	10%	90%	RT_{succ}
10	15	2.4e3	2.7e2	5.5e3	2.4e3	0	*69e+0*	*60e+0*	*88e+0*	1.6e5
1	1	7.4e5	9.4e4	2.0e6	4.4e4	.	.	.	.	.
1e−1	0	*30e−1*	*20e−1*	*50e−1*	1.9e4	.	.	.	.	.
1e−3	.	.	.	.	.	.	.	.	.	.
1e−5	.	.	.	.	.	.	.	.	.	.
1e−8	.	.	.	.	.	.	.	.	.	.

Δf	**f_{5} in 5-D**, N=15, mFE=73					**f_{5} in 20-D**, N=15, mFE=178				
	#	ERT	10%	90%	RT_{succ}	#	ERT	10%	90%	RT_{succ}
10	15	2.4e1	1.0e1	4.6e1	2.4e1	15	1.1e2	8.0e1	1.5e2	1.1e2
1	15	3.5e1	1.3e1	5.4e1	3.5e1	15	1.3e2	9.4e1	1.8e2	1.3e2
1e−1	15	3.6e1	1.3e1	5.6e1	3.6e1	15	1.4e2	1.0e2	1.8e2	1.4e2
1e−3	15	3.6e1	1.3e1	6.7e1	3.6e1	15	1.4e2	1.0e2	1.7e2	1.4e2
1e−5	15	3.6e1	1.7e1	6.7e1	3.6e1	15	1.4e2	9.9e1	1.8e2	1.4e2
1e−8	15	3.6e1	1.3e1	6.7e1	3.6e1	15	1.4e2	9.9e1	1.8e2	1.4e2

Δf	**f_{6} in 5-D**, N=15, mFE=1432					**f_{6} in 20-D**, N=15, mFE=73028				
	#	ERT	10%	90%	RT_{succ}	#	ERT	10%	90%	RT_{succ}
10	15	1.2e2	7.3e1	2.2e2	1.2e2	15	1.7e3	1.4e3	1.9e3	1.7e3
1	15	2.4e2	1.6e2	3.4e2	2.4e2	15	2.6e3	2.1e3	3.2e3	2.6e3
1e−1	15	3.3e2	2.3e2	4.4e2	3.3e2	15	3.5e3	2.6e3	4.4e3	3.5e3
1e−3	15	5.3e2	3.9e2	7.6e2	5.3e2	15	6.5e3	5.1e3	9.3e3	6.5e3
1e−5	15	7.2e2	5.2e2	9.2e2	7.2e2	15	1.1e4	8.4e3	1.5e4	1.1e4
1e−8	15	1.0e3	7.8e2	1.4e3	1.0e3	15	2.6e4	1.3e4	4.1e4	2.6e4

Δf	**f_{7} in 5-D**, N=15, mFE=50004					**f_{7} in 20-D**, N=15, mFE=200004				
	#	ERT	10%	90%	RT_{succ}	#	ERT	10%	90%	RT_{succ}
10	15	9.6e1	3.8e1	2.3e2	9.6e1	15	5.1e4	8.1e3	7.6e4	5.1e4
1	15	6.2e2	1.9e2	1.3e3	6.2e2	0	*62e−1*	*39e−1*	*85e−1*	9.9e4
1e−1	15	4.6e3	8.5e2	6.1e3	4.6e3	.	.	.	.	.
1e−3	11	4.0e4	5.7e3	8.4e4	2.2e4	.	.	.	.	.
1e−5	11	4.0e4	5.7e3	8.3e4	2.2e4	.	.	.	.	.
1e−8	4	1.7e5	3.0e4	4.0e5	2.8e4	.	.	.	.	.

Δf	**f_{8} in 5-D**, N=15, mFE=2977					**f_{8} in 20-D**, N=15, mFE=79401				
	#	ERT	10%	90%	RT_{succ}	#	ERT	10%	90%	RT_{succ}
10	15	1.7e2	6.6e1	4.1e2	1.7e2	15	6.8e3	5.3e3	8.8e3	6.8e3
1	15	8.0e2	2.6e2	2.1e3	8.0e2	15	2.2e4	1.2e4	4.2e4	2.2e4
1e−1	15	1.2e3	6.7e2	1.7e3	1.2e3	15	2.4e4	1.3e4	4.3e4	2.4e4
1e−3	15	1.4e3	1.0e3	2.1e3	1.4e3	15	2.5e4	1.4e4	4.4e4	2.5e4
1e−5	15	1.6e3	1.1e3	2.2e3	1.6e3	15	2.5e4	1.5e4	4.5e4	2.5e4
1e−8	15	1.7e3	1.3e3	2.4e3	1.7e3	15	2.6e4	1.5e4	4.6e4	2.6e4

Δf	**f_{9} in 5-D**, N=15, mFE=5299					**f_{9} in 20-D**, N=15, mFE=35354				
	#	ERT	10%	90%	RT_{succ}	#	ERT	10%	90%	RT_{succ}
10	15	1.9e2	1.0e2	4.4e2	1.9e2	15	5.7e3	5.0e3	6.9e3	5.7e3
1	15	1.3e3	5.0e2	4.4e3	1.3e3	15	1.6e4	1.1e4	2.6e4	1.6e4
1e−1	15	1.6e3	8.6e2	3.1e3	1.6e3	15	1.8e4	1.2e4	3.0e4	1.8e4
1e−3	15	1.9e3	1.2e3	3.4e3	1.9e3	15	1.9e4	1.3e4	3.1e4	1.9e4
1e−5	15	2.0e3	1.3e3	3.4e3	2.0e3	15	1.9e4	1.4e4	3.2e4	1.9e4
1e−8	15	2.2e3	1.4e3	3.5e3	2.2e3	15	2.0e4	1.5e4	3.2e4	2.0e4

Δf	**f_{10} in 5-D**, N=15, mFE=2192					**f_{10} in 20-D**, N=15, mFE=18015				
	#	ERT	10%	90%	RT_{succ}	#	ERT	10%	90%	RT_{succ}
10	15	1.1e3	8.4e2	1.4e3	1.1e3	15	1.4e4	1.2e4	1.6e4	1.4e4
1	15	1.3e3	9.2e2	1.7e3	1.3e3	15	1.5e4	1.4e4	1.6e4	1.5e4
1e−1	15	1.5e3	1.0e3	1.8e3	1.5e3	15	1.6e4	1.5e4	1.7e4	1.6e4
1e−3	15	1.6e3	1.4e3	1.9e3	1.6e3	15	1.6e4	1.6e4	1.7e4	1.6e4
1e−5	15	1.7e3	1.5e3	2.0e3	1.7e3	15	1.7e4	1.6e4	1.7e4	1.7e4
1e−8	15	1.8e3	1.6e3	1.9e3	1.8e3	15	1.7e4	1.7e4	1.8e4	1.7e4

Δf	**f_{11} in 5-D**, N=15, mFE=2359					**f_{11} in 20-D**, N=15, mFE=24413				
	#	ERT	10%	90%	RT_{succ}	#	ERT	10%	90%	RT_{succ}
10	15	1.4e3	1.1e3	1.8e3	1.4e3	15	1.3e4	9.8e3	1.7e4	1.3e4
1	15	1.6e3	1.4e3	1.8e3	1.6e3	15	1.4e4	1.1e4	1.8e4	1.4e4
1e−1	15	1.7e3	1.5e3	1.9e3	1.7e3	15	1.4e4	1.2e4	1.9e4	1.4e4
1e−3	15	1.8e3	1.6e3	2.0e3	1.8e3	15	1.6e4	1.3e4	2.2e4	1.6e4
1e−5	15	1.9e3	1.7e3	2.1e3	1.9e3	15	1.7e4	1.4e4	2.2e4	1.7e4
1e−8	15	2.0e3	1.9e3	2.2e3	2.0e3	15	1.8e4	1.5e4	2.4e4	1.8e4

Δf	**f_{12} in 5-D**, N=15, mFE=12596					**f_{12} in 20-D**, N=15, mFE=42716				
	#	ERT	10%	90%	RT_{succ}	#	ERT	10%	90%	RT_{succ}
10	15	1.6e3	2.0e2	2.9e3	1.6e3	15	5.0e3	1.1e3	1.3e4	5.0e3
1	15	2.6e3	2.7e2	5.2e3	2.6e3	15	9.9e3	1.6e3	2.0e4	9.9e3
1e−1	15	3.3e3	6.9e2	6.5e3	3.3e3	15	1.4e4	7.1e3	2.2e4	1.4e4
1e−3	15	3.9e3	1.3e3	7.4e3	3.9e3	15	1.8e4	1.0e4	2.5e4	1.8e4
1e−5	15	4.6e3	1.3e3	1.1e4	4.6e3	15	2.2e4	1.4e4	3.0e4	2.2e4
1e−8	15	5.1e3	1.6e3	9.4e3	5.1e3	15	2.6e4	1.7e4	3.6e4	2.6e4

Δf	**f_{13} in 5-D**, N=15, mFE=37872					**f_{13} in 20-D**, N=15, mFE=200004				
	#	ERT	10%	90%	RT_{succ}	#	ERT	10%	90%	RT_{succ}
10	15	3.8e2	1.5e2	9.1e2	3.8e2	15	2.9e3	8.5e2	8.4e3	2.9e3
1	15	9.6e2	2.6e2	2.4e3	9.6e2	15	1.3e4	1.3e3	2.4e4	1.3e4
1e−1	15	1.9e3	3.7e2	3.1e3	1.9e3	15	3.0e4	1.0e4	5.6e4	3.0e4
1e−3	15	3.3e3	1.5e3	4.8e3	3.3e3	6	4.0e5	7.0e4	8.2e5	9.5e4
1e−5	15	5.5e3	2.3e3	1.0e4	5.5e3	2	1.4e6	1.5e4	3.4e6	6.1e4
1e−8	15	1.3e4	2.6e3	2.5e4	1.3e4	0	*18e−4*	*24e−7*	*14e−3*	9.2e4

Δf	**f_{14} in 5-D**, N=15, mFE=2618					**f_{14} in 20-D**, N=15, mFE=24006				
	#	ERT	10%	90%	RT_{succ}	#	ERT	10%	90%	RT_{succ}
10	15	1.7e1	4.0e0	5.6e1	1.7e1	15	2.8e2	1.9e2	4.2e2	2.8e2
1	15	7.2e1	4.4e1	1.2e2	7.2e1	15	5.0e2	4.0e2	5.7e2	5.0e2
1e−1	15	1.3e2	9.8e1	1.9e2	1.3e2	15	7.1e2	5.6e2	8.5e2	7.1e2
1e−3	15	4.0e2	2.8e2	5.6e2	4.0e2	15	2.4e3	2.1e3	3.0e3	2.4e3
1e−5	15	1.1e3	9.0e2	1.4e3	1.1e3	15	8.3e3	7.6e3	9.3e3	8.3e3
1e−8	15	2.2e3	2.1e3	2.5e3	2.2e3	15	2.1e4	1.9e4	2.3e4	2.1e4

Δf	**f_{15} in 5-D**, N=15, mFE=50004					**f_{15} in 20-D**, N=15, mFE=200004				
	#	ERT	10%	90%	RT_{succ}	#	ERT	10%	90%	RT_{succ}
10	15	1.4e3	1.8e2	3.4e3	1.4e3	0	*53e+0*	*41e+0*	*63e+0*	8.8e4
1	5	1.2e5	1.2e4	3.3e5	1.9e4	.	.	.	.	.
1e−1	1	7.3e5	7.8e4	1.6e6	2.8e4	.	.	.	.	.
1e−3	1	7.3e5	1.0e5	1.5e6	2.8e4	.	.	.	.	.
1e−5	1	7.3e5	1.3e5	1.6e6	2.9e4	.	.	.	.	.
1e−8	1	7.3e5	7.9e4	1.5e6	2.9e4	.	.	.	.	.

Δf	**f_{16} in 5-D**, N=15, mFE=50004					**f_{16} in 20-D**, N=15, mFE=200003				
	#	ERT	10%	90%	RT_{succ}	#	ERT	10%	90%	RT_{succ}
10	15	6.7e2	6.8e1	1.8e3	6.7e2	15	6.8e4	1.7e4	1.7e5	6.8e4
1	15	9.9e3	3.8e3	1.9e4	9.9e3	0	*55e−1*	*34e−1*	*77e−1*	1.1e5
1e−1	9	5.0e4	7.8e3	1.6e5	1.6e4	.	.	.	.	.
1e−3	2	3.4e5	1.7e4	6.6e5	1.5e4	.	.	.	.	.
1e−5	0	*39e−3*	*54e−5*	*37e−2*	2.1e4	.	.	.	.	.
1e−8	.	.	.	.	.	.	.	.	.	.

Δf	**f_{17} in 5-D**, N=15, mFE=50004					**f_{17} in 20-D**, N=15, mFE=200004				
	#	ERT	10%	90%	RT_{succ}	#	ERT	10%	90%	RT_{succ}
10	15	2.6e1	1.0e0	3.5e1	2.6e1	15	3.8e2	1.9e2	9.4e2	3.8e2
1	15	1.5e2	7.7e1	1.8e2	1.5e2	2	1.4e6	3.1e5	2.8e6	1.2e5
1e−1	15	4.0e3	2.1e2	3.0e4	4.0e3	0	*16e−1*	*98e−2*	*23e−1*	1.2e5
1e−3	0	*12e−3*	*36e−4*	*27e−3*	2.2e4	.	.	.	.	.
1e−5	.	.	.	.	.	.	.	.	.	.
1e−8	.	.	.	.	.	.	.	.	.	.

Δf	**f_{18} in 5-D**, N=15, mFE=50004					**f_{18} in 20-D**, N=15, mFE=200003				
	#	ERT	10%	90%	RT_{succ}	#	ERT	10%	90%	RT_{succ}
10	15	3.2e2	5.1e1	1.8e3	3.2e2	15	2.1e4	9.7e2	6.7e4	2.1e4
1	14	6.9e3	1.6e2	3.0e4	3.3e3	0	*54e−1*	*24e−1*	*91e−1*	7.5e4
1e−1	11	2.5e4	3.6e2	9.3e4	7.2e3	.	.	.	.	.
1e−3	0	*39e−3*	*14e−3*	*50e−2*	1.8e4	.	.	.	.	.
1e−5	.	.	.	.	.	.	.	.	.	.
1e−8	.	.	.	.	.	.	.	.	.	.

Δf	**f_{19} in 5-D**, N=15, mFE=50003					**f_{19} in 20-D**, N=15, mFE=200004				
	#	ERT	10%	90%	RT_{succ}	#	ERT	10%	90%	RT_{succ}
10	15	1.7e1	7.0e0	3.2e1	1.7e1	15	2.3e2	1.0e2	3.5e2	2.3e2
1	15	3.1e3	5.1e2	6.2e3	3.1e3	1	2.8e6	4.4e5	8.1e6	3.9e4
1e−1	5	1.2e5	1.1e4	3.0e5	1.6e4	0	*16e−1*	*11e−1*	*32e−1*	2.0e5
1e−3	1	7.3e5	1.3e5	2.2e6	3.1e4	.	.	.	.	.
1e−5	1	7.3e5	8.1e4	1.5e6	3.1e4	.	.	.	.	.
1e−8	1	7.3e5	8.2e4	1.5e6	3.2e4	.	.	.	.	.

Δf	**f_{20} in 5-D**, N=15, mFE=50004					**f_{20} in 20-D**, N=15, mFE=200004				
	#	ERT	10%	90%	RT_{succ}	#	ERT	10%	90%	RT_{succ}
10	15	4.2e1	9.0e0	6.6e1	4.2e1	15	2.8e2	2.2e2	3.6e2	2.8e2
1	15	6.5e3	2.5e2	1.8e4	6.5e3	5	5.2e5	1.1e5	1.0e6	1.2e5
1e−1	0	*47e−2*	*24e−2*	*87e−2*	2.6e4	0	*11e−1*	*87e−2*	*13e−1*	1.2e5
1e−3	.	.	.	.	.	.	.	.	.	.
1e−5	.	.	.	.	.	.	.	.	.	.
1e−8	.	.	.	.	.	.	.	.	.	.

Δf	**f_{21} in 5-D**, N=15, mFE=15593					**f_{21} in 20-D**, N=15, mFE=81804				
	#	ERT	10%	90%	RT_{succ}	#	ERT	10%	90%	RT_{succ}
10	15	5.1e1	1.1e1	1.1e2	5.1e1	15	1.3e3	2.4e2	3.7e3	1.3e3
1	15	2.0e3	3.8e1	8.5e3	2.0e3	15	1.1e4	5.0e2	3.9e4	1.1e4
1e−1	15	5.2e3	8.9e1	1.5e4	5.2e3	15	2.5e4	2.8e3	5.0e4	2.5e4
1e−3	15	5.3e3	1.6e2	1.5e4	5.3e3	15	2.5e4	3.2e3	5.0e4	2.5e4
1e−5	15	5.3e3	2.1e2	1.5e4	5.3e3	15	2.6e4	3.4e3	4.9e4	2.6e4
1e−8	15	5.4e3	2.7e2	1.0e4	5.4e3	15	2.6e4	3.8e3	5.1e4	2.6e4

Δf	**f_{22} in 5-D**, N=15, mFE=35337					**f_{22} in 20-D**, N=15, mFE=200004				
	#	ERT	10%	90%	RT_{succ}	#	ERT	10%	90%	RT_{succ}
10	15	7.2e2	2.6e1	1.5e3	7.2e2	15	4.2e3	3.2e2	1.2e4	4.2e3
1	15	4.8e3	1.2e2	1.2e4	4.8e3	15	2.8e4	6.9e2	9.3e4	2.8e4
1e−1	15	1.4e4	3.6e2	3.4e4	1.4e4	4	6.5e5	9.3e4	1.5e6	1.0e5
1e−3	15	1.4e4	5.6e2	3.4e4	1.4e4	4	6.6e5	9.4e4	1.4e6	1.1e5
1e−5	15	1.5e4	6.3e2	3.4e4	1.5e4	4	6.6e5	5.9e4	1.4e6	1.1e5
1e−8	15	1.5e4	6.8e2	3.5e4	1.5e4	4	6.6e5	9.6e4	1.2e6	1.1e5

Δf	**f_{23} in 5-D**, N=15, mFE=50004					**f_{23} in 20-D**, N=15, mFE=200004				
	#	ERT	10%	90%	RT_{succ}	#	ERT	10%	90%	RT_{succ}
10	15	6.5e0	1.0e0	1.3e1	6.5e0	15	5.5e1	1.5e0	2.0e2	5.5e1
1	15	4.6e3	4.3e2	9.3e3	4.6e3	10	2.0e5	5.2e4	4.5e5	9.6e4
1e−1	2	3.5e5	3.9e4	1.0e6	2.1e4	0	*66e−2*	*28e−2*	*16e−1*	1.2e5
1e−3	0	*20e−2*	*39e−3*	*35e−2*	2.7e4	.	.	.	.	.
1e−5	.	.	.	.	.	.	.	.	.	.
1e−8	.	.	.	.	.	.	.	.	.	.

Δf	**f_{24} in 5-D**, N=15, mFE=50004					**f_{24} in 20-D**, N=15, mFE=200004				
	#	ERT	10%	90%	RT_{succ}	#	ERT	10%	90%	RT_{succ}
10	15	9.7e3	8.1e2	2.5e4	9.7e3	0	*60e+0*	*49e+0*	*80e+0*	1.1e5
1	1	7.1e5	6.3e4	1.7e6	1.3e4	.	.	.	.	.
1e−1	0	*31e−1*	*11e−1*	*59e−1*	2.5e4	.	.	.	.	.
1e−3	.	.	.	.	.	.	.	.	.	.
1e−5	.	.	.	.	.	.	.	.	.	.
1e−8	.	.	.	.	.	.	.	.	.	.

Table 1: Shown are, for a given target difference to the optimal function value Δf: the number of successful trials (#); the expected running time to surpass $f_{\rm opt}+\Delta f$ (ERT, see Figure 1); the 10%-tile and 90%-tile of the bootstrap distribution of ERT; the average number of function evaluations in successful trials or, if none was successful, as last entry the median number of function evaluations to reach the best function value ($\mathbf{RT_{succ}}$). If $f_{\rm opt}+\Delta f$ was never reached, figures in *italics* denote the best achieved Δf-value of the median trial and the 10% and 90%-tile trial. Furthermore, N denotes the number of trials, and mFE denotes the maximum of number of function evaluations executed in one trial. See Figure 1 for the names of functions.

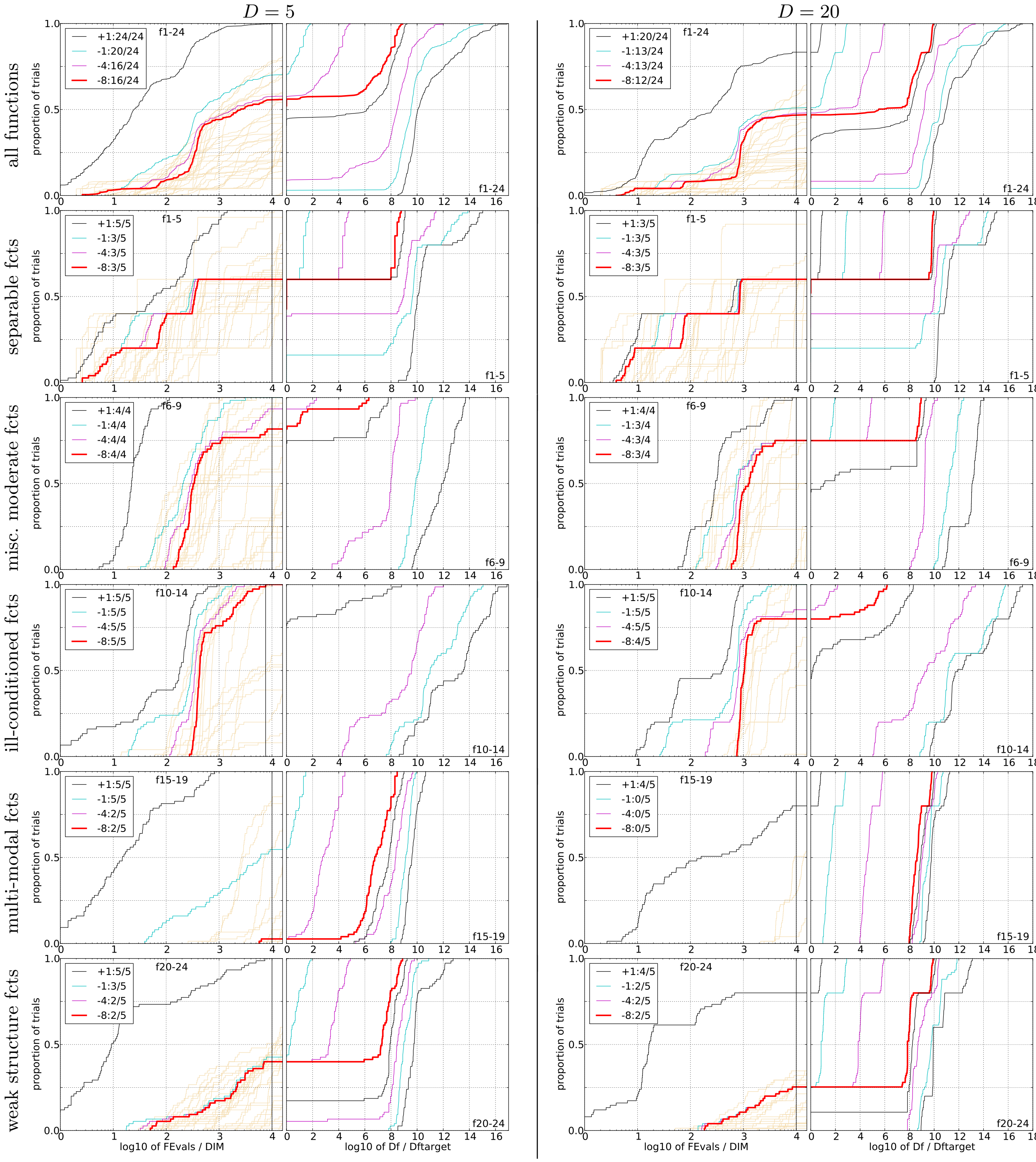

Figure 2: Empirical cumulative distribution functions (ECDFs), plotting the fraction of trials versus running time (left subplots) or versus Δf (right subplots). The thick red line represents the best achieved results. Left subplots: ECDF of the running time (number of function evaluations), divided by search space dimension D, to fall below $f_{\mathrm{opt}} + \Delta f$ with $\Delta f = 10^k$, where k is the first value in the legend. Right subplots: ECDF of the best achieved Δf divided by 10^k (upper left lines in continuation of the left subplot), and best achieved Δf divided by 10^{-8} for running times of $D, 10\,D, 100\,D \ldots$ function evaluations (from right to left cycling black-cyan-magenta). The legends indicate the number of functions that were solved in at least one trial. FEvals denotes number of function evaluations, D and DIM denote search space dimension, and Δf and Df denote the difference to the optimal function value. Light brown lines in the background show ECDFs for target value 10^{-8} of all algorithms benchmarked during BBOB-2009.

(1,4)-CMA-ES on the noiseless BBOB-2010 testbed. In *GECCO (Companion)*, 2010.

[4] A. Auger, D. Brockhoff, and N. Hansen. Investigating the impact of sequential selection in the (1,4)-CMA-ES on the noisy BBOB-2010 testbed. In *GECCO (Companion)*, 2010.

[5] A. Auger, D. Brockhoff, and N. Hansen. Mirrored sampling and sequential selection for evolution strategies. Rapport de Recherche RR-7249, INRIA Saclay—Île-de-France, April 2010.

[6] A. Auger, D. Brockhoff, and N. Hansen. Mirrored variants of the (1,2)-CMA-ES compared on the noiseless BBOB-2010 testbed. In *GECCO (Companion)*, 2010.

[7] A. Auger, D. Brockhoff, and N. Hansen. Mirrored variants of the (1,2)-CMA-ES compared on the noisy BBOB-2010 testbed. In *GECCO (Companion)*, 2010.

[8] A. Auger, D. Brockhoff, and N. Hansen. Mirrored variants of the (1,4)-CMA-ES compared on the noiseless BBOB-2010 testbed. In *GECCO (Companion)*, 2010.

[9] A. Auger, D. Brockhoff, and N. Hansen. Mirrored variants of the (1,4)-CMA-ES compared on the noisy BBOB-2010 testbed. In *GECCO (Companion)*, 2010.

[10] A. Auger, S. Finck, N. Hansen, and R. Ros. BBOB 2009: Comparison tables of all algorithms on all noiseless functions. Technical Report RT-0383, INRIA, April 2010.

[11] A. Auger and N. Hansen. Performance evaluation of an advanced local search evolutionary algorithm. In *Proceedings of the IEEE Congress on Evolutionary Computation (CEC 2005)*, pages 1777–1784, 2005.

[12] A. Auger and N. Hansen. A restart CMA evolution strategy with increasing population size. In *Proc. IEEE Congress On Evolutionary Computation*, pages 1769–1776, 2005.

[13] A. Auger and N. Hansen. Benchmarking the (1+1)-CMA-ES on the BBOB-2009 function testbed. In Rothlauf [28], pages 2459–2466.

[14] A. Auger and N. Hansen. Benchmarking the (1+1)-CMA-ES on the BBOB-2009 noisy testbed. In Rothlauf [28], pages 2467–2472.

[15] P. A. N. Bosman, J. Grahl, and D. Thierens. AMaLGaM IDEAs in noiseless black-box optimization benchmarking. In Rothlauf [28], pages 2247–2254.

[16] S. Finck, N. Hansen, R. Ros, and A. Auger. Real-parameter black-box optimization benchmarking 2009: Presentation of the noiseless functions. Technical Report 2009/20, Research Center PPE, 2009. Updated February 2010.

[17] N. Hansen. The CMA evolution strategy: a comparing review. In J. Lozano, P. Larranaga, I. Inza, and E. Bengoetxea, editors, *Towards a new evolutionary computation. Advances on estimation of distribution algorithms*, pages 75–102. Springer, 2006.

[18] N. Hansen. Benchmarking a BI-population CMA-ES on the BBOB-2009 function testbed. In Rothlauf [28], pages 2389–2396.

[19] N. Hansen, A. Auger, S. Finck, and R. Ros. Real-parameter black-box optimization benchmarking 2010: Experimental setup. Technical Report RR-7215, INRIA, 2010.

[20] N. Hansen, A. Auger, R. Ros, S. Finck, and P. Pošík. Comparing results of 31 algorithms from the black-box optimization benchmarking BBOB-2009. In *Workshop Proceedings of the Genetic and Evolutionary Computation Conference (GECCO 2010)*. ACM Press, 2010. to appear.

[21] N. Hansen, S. Finck, R. Ros, and A. Auger. Real-parameter black-box optimization benchmarking 2009: Noiseless functions definitions. Technical Report RR-6829, INRIA, 2009.

[22] N. Hansen, S. Finck, R. Ros, and A. Auger. Real-parameter black-box optimization benchmarking 2009: Noiseless functions definitions. Technical Report RR-6829, INRIA, 2009. Updated February 2010.

[23] N. Hansen and S. Kern. Evaluating the CMA evolution strategy on multimodal test functions. In X. Yao et al., editors, *Parallel Problem Solving from Nature PPSN VIII*, volume 3242 of *LNCS*, pages 282–291. Springer, 2004.

[24] N. Hansen and A. Ostermeier. Completely derandomized self-adaptation in evolution strategies. *Evolutionary Computation*, 9(2):159–195, 2001.

[25] M. Lunacek, D. Whitley, and A. Sutton. The impact of global structure on search. In *Proceedings of the 10th international conference on Parallel Problem Solving from Nature*, pages 498–507, Berlin, Heidelberg, 2008. Springer-Verlag.

[26] L. Pál, T. Csendes, M. C. Markót, and A. Neumaier. BBO-benchmarking of the GLOBAL method for the noiseless function testbed. `http://www.mat.univie.ac.at/~neum/papers.html`, 2009. P. 986.

[27] R. Ros. Benchmarking the NEWUOA on the BBOB-2009 function testbed. In Rothlauf [28], pages 2421–2428.

[28] F. Rothlauf, editor. *Genetic and Evolutionary Computation Conference, GECCO 2009, Proceedings, Montreal, Québec, Canada, July 8-12, 2009, Companion Material*. ACM, 2009.